CLAIMS

What is claimed is:

1. A cryptography accelerator for generating a stream cipher, the cryptography accelerator comprising:

a key stream generation core for performing key stream generation operations;

a memory associated with the key stream generation core, the memory including a plurality of input ports configured to obtain write data associated with a stream cipher and a plurality of output ports configured to provide read data associated with the stream cipher, wherein the key stream generation core and the memory are operable for performing a plurality of read data operations and a plurality of write data operations associated with generating the stream cipher in a single cycle.

2. The cryptography accelerator of claim 1, wherein generation of the stream cipher is pipelined using coherency checking.

3. The cryptography accelerator of claim 1, wherein the coherency checking comprises determining whether a write address is the same as a read address in a single cycle.

4. The cryptography accelerator of claim 3, wherein a read operation bypasses the memory when the write address is the same as the read address.

5. The cryptography accelerator of claim 1, wherein the stream cipher is associated with three variables.

6. The cryptography accelerator of claim 5, wherein a read operation and a write operation are performed using a first variable and the memory in a first cycle.

7. The cryptography accelerator of claim 6, wherein a read operation and a write operation are performed using a second variable and the memory in a second cycle.

8. The cryptography accelerator of claim 7, wherein a read operation and a write operation are performed using a third variable and the memory in a third cycle.

9. The cryptography accelerator of claim 1, wherein the stream cipher is ARC4.

10. The cryptography accelerator of claim 1, wherein the memory is initialized in a single cycle.

11. The cryptography accelerator of claim 1, further comprising a plurality of byte flops.

12. The cryptography accelerator of claim 1, wherein the key stream generation core is operable to perform key shuffle operations and key stream generation operations.

13. A memory associated with a cryptography engine for generating a stream cipher, the memory comprising:

a plurality of input ports configured to obtain write data associated with generating a stream cipher;

a plurality of output ports configured to provide read data associated with the stream cipher, wherein a plurality of read data operations and the plurality of write data operations associated with generating the stream cipher are performed in a single cycle

14. The memory of claim 13, wherein the stream cipher can be performed in pipelined fashion using coherency checking.

15. The memory of claim 13, wherein the coherency checking comprises determining whether a write address is the same as a read address in a single cycle.

16. The memory of claim 15, wherein a read operation bypasses the memory when the write address is the same as the read address.

17. The memory of claim 13, wherein the stream cipher is associated with three variables.

18. The memory of claim 17, wherein a read operation and a write operation are performed using a first variable and the memory in a first cycle.

19. The memory of claim 18, wherein a read operation and a write operation are performed using a second variable and the memory in a second cycle.

20. The memory of claim 19, wherein a read operation and a write operation are performed using a third variable and the memory in a third cycle.

21. The memory of claim 13, wherein the stream cipher is ARC4.

22. The memory of claim 13, wherein the memory is initialized in a single cycle.

23. The memory of claim 13, further comprising a plurality of byte flops.

24. A method for pipelined generation of a key stream byte, the method comprising:

incrementing a first address during a first clock cycle;

reading a first memory value at the first address, reading a second memory value at the second address obtained by adding the memory value at the first address to a previous second address, writing the first memory value to the second address and the second memory value to the first address, and summing the first and second memory values to yield a third address during a second clock cycle;

reading a third memory value at the third address during a third clock cycle.

25. The method of claim 24, further comprising performing read-after-write coherency checking.

26. The method of claim 25, wherein read-after-write coherency checking comprises determining whether a first memory value at a first address is being read and written in the same clock cycle.

27. The method of claim 25, wherein the given memory value is bypassed by a read operation if the first address is being read and written in the same clock cycle.

28. The method of claim 24, further comprising performing write-after-write coherency checking.

29. The method of claim 28, wherein write-after-write coherency checking comprises determining whether a first address is being written to twice in the same clock cycle.

30. The method of claim 28, wherein a single write is performed if it is determined that a first address is being written to twice.

31. The method of claim 24, further comprising providing the third memory value as a key stream byte.

32. The method of claim 31, wherein the key stream byte is an ARC4 key stream byte.

33. The method of claim 24, further comprising initializing the memory in a single cycle.

34. The method of claim 33, wherein initializing the memory comprises placing the memory address into the corresponding memory value.

35. The method of claim 34, wherein the key stream byte is associated with the generation of a multibyte ARC4 key for encrypting a data stream.

36. The method of claim 34, wherein the key stream byte is associated with the generation of a multibyte ARC4 key for decrypting a data stream.

37. A cryptography accelerator for pipelined generation of a key stream byte, the cryptography accelerator comprising:

means for incrementing a first address during a first clock cycle;

means for reading a first memory value at the first address, reading a second memory value at the second address obtained by adding the memory value at the first address to a previous second address, writing the first memory value to the second address and the second memory value to the first address, and summing the first and second memory values to yield a third address during a second clock cycle;

means for reading a third memory value at the third address during a third clock cycle.

38. The cryptography accelerator of claim 37, further comprising means for performing read-after-write coherency checking.

39. The cryptography accelerator of claim 38, wherein means for read-after-write coherency checking comprises means for determining whether a first memory value at a first address is being read and written in the same clock cycle.

40. The cryptography accelerator of claim 38, wherein the given memory value is bypassed by a read operation if the first address is being read and written in the same clock cycle.

41. The cryptography accelerator of claim 37, further comprising means for performing write-after-write coherency checking.

42. The cryptography accelerator of claim 41, wherein write-after-write coherency checking comprises means for determining whether a first address is being written to twice in the same clock cycle.

43. The cryptography accelerator of claim 41, wherein a single write is performed if it is determined that a first address is being written to twice.

44. The cryptography accelerator of claim 37, further comprising means for providing the third memory value as a key stream byte.

45. The cryptography accelerator of claim 44, wherein the key stream byte is an ARC4 key stream byte.